

## REMARKS

The above amendment and these remarks are in response to the Office Action mailed 01/30/2003 by Examiner Mareisha N. Winters.

Claims 1-19 are in the case, none having as yet been allowed.

### *Oath/Declaration*

The oath or declaration has been objected to as defective for not identifying the citizenship of Carl J. Kraenzel.

Applicants submit herewith a copy of the corrected declaration, the original of which has been corrected to show the citizenship of Carl J. Kraenzel, the correction initialed, and signed and dated anew by Carl J. Kraenzel.

Applicants request that the corrected declaration be accepted.

### ***Drawings***

Informal drawings filed with the application have been accepted for examination purposes only.

Applicants submit herewith a letter to the drawing review branch, together with formal drawings for entry in the case subject to the approval of the Examiner.

### ***Specification***

The specification has been corrected at pages 1 and 2 to provide appropriate serial numbers and filing dates for the related applications.

### ***Claim Objections***

Claims 7 and 17 have been objected to because of spelling informalities.

Applicants have amended claims 7 and 17 to correct the misspelled words.

**35 U.S.C. 102**

Claims 1-19 have been rejected under 35 U.S.C. 102(e) over Jerger et al. (U.S. Patent 6,473,800).

Applicants have provided a system and method which differs from Jerger in at least the following aspects:

1. Centrally managed security policy.
2. Non-conflicting, wholly separate execution spaces.

**1. Centrally managed security policy.**

Jerger, which describes Microsoft Corporation technology, does not teach any security-policy delegation. In Jerger, all trust and enablement decisions are in the hands of the baffled end-user. Most end users are not sophisticated enough to proactively and dynamically manage all the browser choices on zones, prompts, and so forth. Indeed most don't even understand the basic ActiveX install dialog and either (a) fearfully don't accept safe installations because the signature name has no apparent

relationship to the site they are at (e.g., signed by Microsoft but looking at an Oracle site) or (b) carelessly accept unsafe installations because it looks signed by somebody. The result is many corporations simply have established a requirement that end users disable all ActiveXs.

Applicants model, by contrast, allows the user to make one trust decision only. That is, does the user trust the administrator of the site being accessed? If the user extends that trust, then all finer-grained decisions on what agents can or can't run locally, future updates that bring additional agents or tune the policies of what is allowed to run, and so forth ... all that management is done by the site administrator, not the end user. Centrally managed policy for what has rights to run, and how, on the end user's desktop is provided by Applicants but is fundamentally lacking in the Jerger (Microsoft) model.

In their specification, applicants describe this aspect of their invention with respect to Figures 13 and 26. First, the user is queried to determine if the site is to be trusted. This is described as follows:

"...to execute custom code install from the server to the client. This involves the creation of a permission moment, a moment in which the user is prompted to respond to two queries: (1) site identity: does the user believe that the server is who it represents itself to be; and (2) site trust: does the user trust the server to place the custom code on the client machine."

"Referring to Figure 26, in accordance with the preferred embodiment of the invention, site identity is associated with the secure sockets (SSL) signature, and whether the connection to the web site has been made using HTTPS (secure) or HTTP (not secure). If in step 553 it is determined that the user has connected to the server web site in step 551 using SSL, then the site identity and site trust queries are presented by stating (1) in step 555, the site has been verified as being what it represents itself to be, and (2) in step 556, asking "Do you trust the web site to download custom code to your client machine?" If the user has not connected to the server web site using SSL, then the site identity and site trust queries are presented by stating (1) in step 554, the site has not been

verified as being what it represents itself to be, and  
(2) again in step 556, asking if the web site is to be  
trusted to download custom code to the client machine?"

[Specification, page 133, line 7 to page 134, line 8.  
Emphasis added.]

Thereafter, code is downloaded from the server based upon  
user acceptance of the web site. This is described as  
follows:

"Download of the custom code proceeds based upon the  
user determination in step 557 that the web site,  
whether verified or not, is to be trusted.

[Specification, page 134, lines 9-11. Emphasis added.]

As noted above, once the user accepts downloads, from  
that point on things are pushed down from the server. This  
is further illustrated in Fig. 13, which shows how cross  
certificates 566, restricted list 570, unrestricted lists  
573 are pushed down from server 100 to client 200 to affect  
what runs in a subscription 201 -- these operations  
occurring well after the user 200 has accepted site 100 as a  
trusted site.

Applicants have amended the independent claims 1, 12, 18 and 19 to make clear that once a site is accepted as trusted, security policies are centrally administered by the system administrator.

**2. Non-conflicting, wholly separate execution spaces.**

The Jerger model does not allow for multiple versions of an ActiveX to exist simultaneously on a client machine, each executing in wholly separate execution spaces under different security policies, all unaware of each other and non-conflicting.

On the other hand, applicants' model does keep the security context of downloaded active content in completely separate spaces, governed only by policies that come from the separates source sites: the sites are fundamentally unable to affect each other's components even if they are literally the same components.

For example, if two different travel websites utilize and ActiveX signed by Mapquest according to Verisign, if travel site A upgrades ists ActiveX, then end users that use both sites would be forced into using the most recent

version, even if this might break travel site B. Nor does one travel site have the ability to protect its use of the ActiveX only to its won context by setting policies on it and blocking it from third-party use.

With applicants' model, both travel sites could have the same (or even same with version deltas) signed agent that came from Mapquest, and be guaranteed that those agents would run in private non-interacting secure spaces on the end user machine, spaces that can even be encrypted.

This aspect of applicants' invention may be illustrated with reference to Figure 13. In Figure 13, signed agents 562, 563 can be the exact same agent code (for example, "out of office agent, published/signed by IBM"), but of slightly different versions (e.g. 1.1 vs 1.2). Figure 13 shows that client 200 has separate execution spaces for subscriptions 202 and 201, with separate agents 562, 563 within each subscription. Indeed, even the policy is separate... that is, server B 101 could choose, for example, to move this agent 561 from its unrestricted list 574 to its restricted list 571, and that would only impact its offline version. Server B 101 has no way of affecting the files from server A 100, or even being aware of them. Figure 13 is described in

applicants' specification as follows:

" Referring to Figure 13, system components exercised in qualifying signed agents 560, 561 from a plurality of servers 100, 101 for execution as signed agents 562, 563 at a client 200 are illustrated. Server directors 350, 351 include certificates 564, 565, cross certificates 566, 567, downloadable cross certificates 568, 569, restricted group lists 570, 571 and unrestricted group lists 573, 574. Client 200 includes client side rendition 202, 562 of application 136 with signed agent(s) 560, and client side rendition 201, 563 of server application 137 with signed agent(s) 561; and client directory 212 with downloadable cross certificates 576, unionized restricted group list 572 and unionized unrestricted group list 575. A signature is a name plus an electronic certificate. Group lists 572, 575 include names, not complete signatures. Unionized group lists include the union of names 570, 571 and 573, 574 from all servers, in this example two servers 100 and 101 are shown, but there may be more." [Specification, page 44, line 11 to page 45, line 6. Emphasis added.]

As will be apparent to those of skill in the art, it is inherent that unionized group lists 572, 575 imply that servers A and B are unaware of each other, and to do a union, agents 562 and 563 must run in separate partitions.

There is nothing in the Jerger (Microsoft) model that allows off-line enabled, completely partitioned execution spaces, where the security policy of what happens in those spaces can be centrally administered. Microsoft JVMs and applets are the closest thing, in that applets from server A vs server B can run in partitions, but these have no offline-ability built into them. As such, the applets will run when the user is connected, but the site has no meaningful generic way to push its data to a desktop, along with processing logic and a snapshot of security policies so that this will all run seamlessly the same way when the user is disconnected.

Applicants have amended the independent claims 1, 12, 18 and 19 to clarify that simultaneously existing agents execute in separate, non-conflicting execution spaces.

## SUMMARY AND CONCLUSION

Applicants urge that the above amendments be entered and the case passed to issue with claims 1-19.

If, in the opinion of the Examiner, a telephone conversation with applicant(s) attorney could possibly facilitate prosecution of the case, he may be reached at the number noted below.

Sincerely,

Carl J. Kraenzel, et al.

By

  
Shelley M Beckstrand  
Reg. No. 24,886

Date: 30 April 2003

Shelley M Beckstrand, P.C.  
Attorney at Law  
314 Main Street  
Owego, NY 13827

Phone: (607) 687-9913  
Fax: (607) 687-7848